This application is submitted in the name of inventors Darrell Shively, John Knight, Kavita Patil, Pauline Chen Boyd, Sonny Bui, and Thomas Roden, all assignors to Cisco Technology, Inc., a California Corporation.

SPECIFICATION

TITLE OF INVENTION

AUTOMATIC HARDWARE FAILURE DETECTION AND RECOVERY FOR DISTRIBUTED MAX SESSIONS SERVER

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to the field of data communications networks. More particularly, this invention relates to a method and apparatus for automatically detecting hardware and communication failures and accordingly adjusting the true count of users logged into a Max Session Server (MSS). The invention also has applicability to other forms of resource management within a data communications network.

The Background

A user, or subscriber, of a network system can remotely log into a data communications network and access resources, such as the Internet, provided by the server. Both businesses and individuals can be users or subscribers. The network systems are typically operated by Internet Service Providers (ISPs),

telephone companies, or Online Service Providers (collectively referred to as ISPs). There are numerous transmission media available to connect to the ISPs, including dialing in over the telephone network (PSTN) or connecting in another conventional manner such as via DSL (digital subscriber line), cable, ISDN (integrated services digital network), etc. Via whichever selected form of transmission, users typically gain remote access through a network access server (NAS). The NAS then requires some type of unique identification to allow access, such as a user name and password.

FIG. 1 is a diagram depicting a typical relationship between users and the server. The users (clients or subscribers) can log into a number of network access servers (NAS_1, NAS_2 and NAS_3), which provide data communications portals to a point of presence (PoP) on the data communications network. Each NAS is in communication with a conventional AAA (authentication, authorization and accounting) or similar service to determine if the log in is authorized. If authorized, the user then gains access to the network service.

Most ISPs provide large numbers of NASs to allow numerous users at various geographic regions to gain access to the system. However, it often becomes necessary to keep track of the total number of users or groups of users logged into the multiple NASs. For example, a company may purchase access for fifty of its employees at any given time from an ISP. Thus, the ISP needs to keep

track of how many users from the particular company are logged into the system. Similarly, a single user may only pay for access to the system from one connection at any given time. However, a home user with multiple computers could attempt to log in from several computers. It is in the interest of the ISP to limit that user to only the one session that the user has purchased.

In order to keep track of the number of log ins, ISPs or Online Services may utilize a Max Sessions Server (MSS), which can either be a separate entity or integrated with an Authorization, Authentication and Accounting server (AAA) and is commercially available from vendors such as Cisco Systems, Inc. of San Jose, California. The MSS restricts a user or a group of users (collectively referred to as a group) to a maximum number of sessions across a complete administrative domain. It does this by maintaining a counter for each user or group of users. A single user may belong to multiple groups, where each group has its own session counter. For each logged in user added, the corresponding counter(s) is incremented by one. In the event that a user belongs to multiple groups, the counter for each associated group will be incremented. For example, a company may allocate 200 logins for the engineering group, which may be further subdivided into 50 logins for hardware engineering group, 50 logins for the systems engineering group, and 100 logins for the design engineering group. When a user belonging to the systems engineering group logs in, the counter for both the systems engineering group and the overall engineering group will be

incremented by one. When the user logs out of the NAS, the NAS sends an accounting record to the AAA server with a conventional protocol such as RADIUS or TACACS+ indicating that the session has stopped. The AAA server notifies the MSS that the user at a particular NAS and port has logged off and the associated counter(s) for that user are decremented by one.

FIG. 2 is a flow diagram of the communication between client and server. The user connects to a NAS, which then sends a request for authorization to the AAA. The AAA sends a request to a Max Sessions Server (MSS) to determine if there are available slots left for the user to log into the system. If the connection is within the allotted number of log ins for that user or group of users, then the request is granted and the corresponding counter is incremented by one. However, if the connection would result in more log ins than are allotted to the user or group, then the request is denied.

It is important to note that each MSS maintains a counter for a particularly designated user or group of users and only that MSS will maintain the count for that designated user or group of users. For example, a company may have a systems division that has 200 logins allotted and a hardware division that has 200 logins allotted. One MSS may maintain the counter for both of these groups or there may be two MSSs, where one handles the systems division and one handles the hardware division.

Consider what happens when a user ($USER_{A10}$) belonging to a group at site A of a company travels to site B of the company. Referring to FIG. 3, each site at the company has an MSS, which maintains its list of authorized number of users per group at each respective site. When $USER_{A10}$ attempts to log into the server at site B, through the log in process the MSS_B identifies the user's group and then recognizes that it does not maintain the counter for the user's group. Instead, it will proxy the request to MSS_A. Assuming the user is authorized to log in, the user will be located at a port on the NAS at site B, but accounted for at the MSS at site A. In other words, MSS_A will add the connection to $USER_{A10}$ to its count for users belonging to group A.

When a hardware or communication failure occurs, a user or group of users may actually be logged out through disconnection (abnormally disconnected) and yet the MSS will not be notified. Therefore, the count of the number of sessions maintained by MSS for the user or group will be more than the actually existing number of sessions. The MSS may deny users access based on the inaccurate count, when the users should be granted access. This will result in the user or group receiving fewer connections than entitled, which is a condition known as "under-subscription." ISPs do not want to create customer dissatisfaction; therefore, this result is highly undesirable.

What is needed is an addition to the present MSS that can automatically detect hardware and communications failures and adjust the session count accordingly. This would overcome the under-subscription problem by allowing the correct number of users authorized on the system to log in.

SUMMARY OF THE INVENTION

A failure detection system operates in conjunction with a Max Sessions Server. A first type of failure detection operates at the user level and is implemented every time a new user attempts to log into the system. Upon log in, the user is assigned to a particular NAS and port. The NAS sends a request to an AAA for authorization, which in turn queries the MSS to determine if the log in will violate the allotment for that particular user or group of users. The MSS will only allow a predetermined number of connections at a time for each user or group of users. A master list of unique identification values (UIVs) for all logged in users is maintained at the MSS. The UIV may be a concatenation of the NAS identification and port identification and should be unique to it. Since only one connection can be made on a given NAS at a given port, this number is unique at any given time. For each new AAA request, the MSS compares UIV of the new request with the UIVs already in use in the MSS master list. If the UIV is duplicated, then the system can correctly conclude that a hardware or communications failure occurred and the original user at that NAS/port has been disconnected. In this case, the system will decrement the appropriate MSS counter(s) by one and remove the UIV from the master list. The login request will then proceed as normal in an MSS request, where the AAA request will only be granted if there is space available in the allocated number of slots for the user or group of users.

A second type of failure detection operates at the NAS level. When a particular NAS has failed to communicate with the MSS for a given length of time, the system concludes that there is a communications or hardware failure on that NAS. All of the UIVs associated with that particular NAS are removed from the master list and the corresponding counter for the MSS is decreased by the total number of users that were previously logged into the failed NAS. Furthermore, since multiple MSS systems may be associated with a given NAS, a message is broadcast to all previously interested MSSs to remove the associated UIVs and decrement their counters. This broadcast is performed by sending data packets to the interested remote MSSs over the net encapsulated in TCP (Transmission Control Protocol).

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram showing a simple client/server relationship system in accordance with the prior art.

FIG. 2 is a flow diagram of the operation of a Max Sessions Server (MSS) in accordance with the prior art.

FIG. 3 is a block diagram showing a proxy request between two MSS at different sites in accordance with the prior art.

FIG. 4 is a flow diagram illustrating the operation of the failure detection system for a user level failure in accordance with a presently preferred embodiment of the present invention.

FIG. 5 is a flow diagram illustrating the operation of the failure detection system for a NAS level failure in accordance with a presently preferred embodiment of the present invention.

FIG. 6 is a system block diagram showing a system in accordance with a presently preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Those of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled persons having the benefit of this disclosure.

In accordance with a presently preferred embodiment of the present invention, the components, processes and/or data structures may be implemented using C++ programs running on high performance computers (such as an Enterprise 2000™ server running Sun Solaris ™ as its operating system). The Enterprise 2000™ server and Sun Solaris™ operating system are products available from Sun Microsystems, Inc. of Mountain View, California. Different implementations may be used and may include other types of operating systems, computing platforms, computer programs, firmware and/or general purpose machines. In addition, those of ordinary skill in the art will readily recognize that devices of a less general purpose nature, such as hardwired devices, devices relying on FPGA (field programmable gate array) or ASIC (Application Specific Integrated Circuit) technology, or the like, may also be used without departing from the scope and spirit of the inventive concepts disclosed herein.

This invention teaches two levels of automatic failure detection and correction. The first level of detection operates to detect user-level failures. The second level of detection operates to detect NAS-level failures. In order to implement both levels of detection, the MSS must maintain a master list of UIVs for each user log in. To create the UIV, the MSS creates a concatenation of a unique NAS identifier (either its IP address or "name") and the port identifier of the port of the NAS through which the user is connected. Since only one user can be logged into a particular port on a particular NAS, this number is unique to the connection. Furthermore, all UIVs associated with a given NAS are readily identifiable.

As an example, assume the NAS IP address is 10.1.1.10 and the NAS name is MYNAS and assume the user logs into the NAS at port TTY2. Either the NAS IP address or name can be used in the concatenation. Thus, the UIV could be either 10.1.1.10#TTY2 or MYNAS#TTY2 or another suitable UIV formed in this general way. However, one protocol would be selected and followed consistently. In other words, the MSS would not sometimes use the NAS IP address and sometimes use the NAS name. Similarly, the order of the concatenation does not matter (either NAS first or Port first); however, whichever convention is selected would always be used for consistency.

Referring to FIG. 4, the user-level failure detection 12 is diagramed. When the user logs into the NAS at a particular port, the NAS then sends an access request packet to the AAA (or an equivalent authorization server) for authorization. The AAA must then request permission from the MSS to allow an additional user to log into the system. Assuming that the MSS maintains the counter for the user or user's group, the MSS will follow multiple steps. (Note that in the event that it does not, the request will be proxied to the appropriate MSS and that MSS will follow the steps below.) First, it will generate a UIV based on the particular port and NAS to which the connection is logged in. Then, the MSS will compare (block 14) this the UIV of the requesting user to the master list of UIVs that are already logged into the system. If the UIV is not on the master list, then there has been no user level failure. At this point, the MSS proceeds to verify if the maximum number of log ins for the user or user's group would be violated if the AAA request is granted (block 16). If the maximum would be exceeded, the request is denied with an access-reject packet (block 18). If the maximum would not be exceeded, then the request is granted with an access-accept packet (block 20). Once authorized, the MSS then increments the corresponding counters by one (block 22) and adds the UIV to its master list (block 24).

If however, when the MSS compares the UIV of the current log in request to the master list, the number is found on the list, then the MSS concludes that a communication or hardware failure occurred. Since it is impossible for two

different users to log in with the same UIV (based on NAS and port number), then the older connection must have been lost due to a hardware or communications failure allowing a second user to connect to the NAS and port number of the prior user's connection. In this case, the MSS takes several steps. The MSS removes the UIV from the master list (block 28) and decrements the corresponding counters by one (block 26). At this point, the over-estimation has been corrected and the system will proceed with the authorization request. Then, the MSS will consult the corresponding counter to see if the maximum number of users would be exceeded if the AAA request is granted. If so, then the request is denied; if not, then the user is authorized. Once authorized, the MSS will add the newly logged in UIV to the master list and increment the corresponding counter(s) by one. Thus, no connection has been denied based on an inaccurate log in count from user level failures. Note that NAS-AAA-MSS communications may be carried out in any suitable fashion, such as by use of the well-known RADIUS or TACACS+ protocols.

The second level of failure detection is a NAS-level failure and is diagrammed at FIG. 5. If the NAS goes down or fails to communicate with the MSS (block 30) within a predetermined amount of time, the MSS concludes that the NAS has had a hardware or communications failure and has become entirely inoperative. In this type of communications or hardware failure, potentially

numerous connections from users have been terminated but are still recorded as being logged in. The MSS count is over-estimated and must be corrected.

According to the present invention, the MSS will automatically correct for the lost connections from a NAS-level failure. Referring to FIG. 5, if the NAS has failed to communicate, then all UIVs associated with the NAS will be removed from the master list (block 32). As described above, the UIVs are a concatenation of the NAS and port identifiers; therefore, the MSS will be able to easily determine which UIVs are associated with the failed NAS. The MSS will adjust the corresponding counters by the number of lost sessions 34.

Additionally, there can be multiple MSSs associated with users on the failed NAS. Recall the scenario of FIG. 3, where a user belonging to a group at site A of a company travels to site B of a company. When the user attempts to log into the server at site B, the MSS_B will proxy the request to MSS_A. Assuming that the user is authorized to log in, the user will be located at a port on the NAS at site B, but accounted for at the MSS at site A. If the NAS_B crashes, then the MSS_B will detect the failure, and it will adjust accordingly for all lost connections for which it maintains the counters and master identification lists. However, the MSS at site A will not know about the failure at the NAS at site B and will be under-subscribed since it still has a user counted for that is no longer connected. Therefore, in this situation, it is preferred that the MSS_B broadcast a message to

MSS_A to notify it of the failure. In order to notify MSS_A a suitable data packet is sent over the data communications network encapsulated in a suitable protocol such as TCP (transmission control protocol).

When a NAS failure is detected, the MSS will identify any remote MSS that has previously shown an interest in a connection on the failed NAS and will broadcast the information. Thus, all counts and master lists on other MSSs will also accurately reflect the result of a failed NAS.

In an extension of the present invention, a more generic Resource Control Server (RCS) could operate in the same basic way as the MSS described above to control the allocation of resources other than sessions. Such resource could include any type of limited resource within the data communications network, such as, for example, call gateways, VPNs (virtual private networks), B-channels (used with ISDN connections), and the like. Members of groups would subscribe for minimum service levels of the resource and when the minimum service level is provided, no further service need be provided. The same problem of under-subscription still exists and can be corrected with the present invention.

FIG. 6 shows a system block diagram of a system in accordance with a presently preferred embodiment of the present invention.

A MSS or RCS 40 includes a database 42 storing the master list 44 which stores UIVs and associated group identifications. The MSS/RCS 40 also stores and maintains counters for each group ID 46. A checker 48 compares each new log in request directed to MSS 40 with the contents of master list 44 to determine if the UIV of the new log in request matches an existing UIV in the master list 44 of database 42. A clearer 50 clears existing information in master list 44 associated with a UIV if the UIV is determined by the checker 48 to be the same as that of the new log in request. An incrementer 52 increments a counter 46 for each new log in by a member of a corresponding group of users. For the RCS version, this occurs when a user initiates use of the controlled resource. A decrementer 54 decrements a counter 46 when the user logs out or gives up the resource. A log in rejector 56 rejects a user's attempt to log in (MSS version) or gain access to a resource (RCS version) if doing so would cause the corresponding counter to exceed its authorized maximum. A NAS checker 58 periodically checks one or more associated NASs to determine if it/they have become non-operational. Note that this function may be implemented in the MSS/RCS or in another portion of the data communication network such as a AAA server or a network operations center (NOC) with the results of the check or a failure notification sent to the MSS/RCS. A broken NAS clearer 60 responsive clears existing data in the master list associated with a broken or non-operational NAS. A transmitter 62 transmits a communication to another MSS on the data communications network to inform it of the non-operational status of a NAS. A

receiver 64 receives communications from a transmitter of another MSS or another source informing of the non-operational status of a NAS. Such information may be passed to broken NAS clearer 60 for action.

An AAA 66 couples the NASs 68, 70, 72 with the MSS 40 via a suitable protocol such as RADIUS or TACACS+.

Alternative Embodiments

While embodiments and applications of the invention have been shown and described, it would be apparent to those of ordinary skill in the art having the benefit of this disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts herein. A more generic Resource Control Server (RCS) could operate in the same basic way as the MSS described above to control the allocation of resources other than sessions. Such resource could include any type of limited resource within the data communications network, such as, for example, call gateways, VPNs (virtual private networks), B-channels (used with ISDN connections), and the like. Members of groups would subscribe for minimum service levels of the resource and when the minimum service level is provided, no further service need be provided. The same problem of under-subscription still exists and can be corrected with the present invention. The invention, therefore, is not to be restricted except in the spirit of the appended claims.